J O I N T C E N T E R
AEI-BROOKINGS JOINT CENTER FOR REGULATORY STUDIES

Evaluating Total Cost of Ownership for Software Platforms:

Comparing Apples, Oranges and Cucumbers

Alan MacCormack

Related Publication 03-9

April 2003

Alan MacCormack is the Lumry Family Assistant Professor of Business Administration at the Harvard Business School. The author gratefully thanks Roy Levien, Bernard Reddy and James Hunter of National Economic Research Associates (NERA) for providing research assistance in this work. Funding for this paper was provided in part by Microsoft. The opinions expressed in this paper are solely those of the author. All correspondence should be addressed to the author at amaccormack@hbs.edu.

Executive Summary

This paper reports findings from a review of articles and studies that examine the Total Cost of Ownership (TCO) of different software platforms. In particular, we assess the "state-of-practice" in measuring TCO, through a comparison of the methods employed and data provided in studies that investigate the costs of competing software operating systems. Our results suggest that despite increasing interest in this method of evaluating different software platforms, many studies suffer from a lack of reliable data on which to base conclusions, as well as a lack of consistency in the approach taken to calculate TCO. Furthermore, only a handful of studies gather data from a large enough sample of firms to inspire any confidence that the results are representative of a broader population. While some organizations (e.g., Gartner and IDC) have made notable efforts to overcome these problems, more work is needed to ensure greater consistency across studies.

Despite the problems noted, several important conclusions *can* be drawn from a review of the more comprehensive studies completed to date. First, acquisition costs for software tend to be dwarfed by other costs, typically comprising less than 10% of the TCO for a system. This suggests that whether software is free, cheap or relatively expensive has relatively little impact on the total cost of Information Technology (IT) investments. By contrast, the single largest component of cost is staffing, typically comprising 50-70% of the TCO for a system. This suggests that TCO studies should expend significant amounts of effort on assessing the drivers of differences in staffing cost across the systems under examination.

At a broader level, our work suggests a need to realize the limitations of a TCO-based metric, given this represents only one dimension of the IT investment decision. In particular, TCO studies typically assume that different software platforms deliver the same value to users, an assumption that is not always valid. For example, software platforms may differ in both the functionality that they provide to users and the breadth of applications that a user has access to. In addition to Total Cost of Ownership, a robust IT investment decision must therefore consider the Value Delivered to Users, integrating both these factors into a forecast Return on Investment from planned IT expenditures.

Evaluating Total Cost of Ownership for Software Platforms: Comparing Apples, Oranges and Cucumbers

Alan MacCormack

Introduction

There has been increasing interest in recent years in calculating the costs of ownership for various software platforms, with the aim of helping CIOs make better decisions as they purchase, upgrade and/or replace their IT infrastructures. Correspondingly, there has been significant growth in the number of "studies" which aim to provide information on the comparative costs of such platforms over their useful life – a figure referred to as the Total Cost of Ownership (TCO). Unfortunately, many of these studies suffer from a lack of reliable data on which to base more general conclusions, as well as a lack of consistency in the approach taken to calculating TCO. This paper addresses these concerns through an assessment of the state of practice in measuring TCO.

The paper summarizes some of the major problems that are evident in existing attempts to calculate and compare TCO figures for different types of software platform. The work is based upon a review of more than 80 articles and studies discussing the life cycle costs for a specific type of platform – the server operating system. The intention of this paper is not to be critical of studies that bring reliable data to bear on the topic of TCO – indeed, such studies are to be commended, given many efforts rely upon only anecdotal evidence, dubious assumptions about operating environments, or informed speculation. Rather the aim is to highlight deficiencies in the state of practice for measuring TCO, with a call for participants in the industry to work for greater consistency in how these studies are conducted. The paper also highlights some of the more general conclusions that *can* be drawn from the review of articles and studies compiled in this work.

Background

No firm buys an operating system purely for its own sake. Operating systems are only valuable as a platform upon which other, more useful, applications can be run –

applications such as web, file and print serving; email, calendaring and groupware; and database and network management. Any evaluation of TCO for an operating system software platform is therefore context specific. It will include not only the cost of purchase and maintenance of the operating system, but also the purchase and maintenance of the applications that run on top of this system, and the hardware that sits below it. Market data on server operating systems suggest that different firms can and do reach different conclusions concerning their choice of software for these various applications (Table 1). This conclusion is supported by the heterogeneity apparent in a review of the many different articles and studies written on TCO (see below). Such studies typically find that the TCO for a software platform is highly dependent on the type of workload being assessed (e.g., web versus file servers) as well as other contextual factors such as the size and nature of the firm within which this workload resides.

Table 1: Worldwide Shipments of New Servers, by Operating System, 2002

Operating System	*Revenues (USD Millions)*	*Units*
Unix	$17,152	609,953
Windows NT	$12,108	2,707,158
NetWare	$1,717	437,158
Linux	$2,018	597,606
OS/400	$1,913	21,571
OS/390	$3,638	1,873
Other	$3,345	58,109
Total	$41,891	4,433,429

Source: IDC Server Tracker Database, Q4 2002, excluding upgrade shipments.

The State-of-Practice in Evaluating TCO

In order to get a sense of the state of practice in evaluating TCO, we identified and reviewed as many publicly available articles and studies on TCO for different server operating systems that we could find. We focused on this particular type of software platform because there has been a wealth of recent articles written about TCO for operating systems, spurred to a large degree by the rise of Linux, an “open source” software product which increasingly competes head-to-head with both proprietary Unix systems and with Microsoft’s Windows software.

Our search was conducted using a variety of sources, including web searches, reports from IT consulting firms such as Gartner Group, IDC and Meta Group, and case studies or other data provided by solution providers themselves (e.g., by Microsoft). This process identified 84 articles and studies that reported some information (if only background and concepts) about TCO for different software operating systems (we have listed these articles and studies in the appendix). Note that our search cannot be regarded as exhaustive, given, for example, that there are many studies of software TCO that are proprietary to the organizations that conducted or funded them. This sample is therefore biased to the degree that these studies differ to the ones that are publicly available.

After reviewing the articles and studies in this sample, we eliminated 29 of these because they provided no primary source material. We defined primary source data in a broad way, to include evidence on actual purchase or maintenance costs for various IT configurations, data from specific IT implementations (whether gathered in a systematic way or not) and articles which came to conclusions based upon what we call "informed speculation" – in essence, calculating TCO figures based upon assumptions (rather than data) about the costs involved in deploying alternative platforms. This subset of 55 articles provided the basis for our review.

It is important to point out that our objective in reviewing this sample was not to attempt to draw conclusions about the relative merits of the specific software platforms that these studies examined. Our review was aimed instead at assessing the data and methods by which these studies arrived at their conclusions. Indeed, given our findings in this latter respect, we believe it would be premature to declare any specific platform a victor in TCO for any particular type of workload or context. The inconsistencies in methods and the conflicting data on costs that are apparent when reading these studies are unfortunately large enough to make the drawing of such conclusions unwarranted. We describe the reasons for coming to this conclusion below.

1: Many studies provide inadequate data with which to support their conclusions.

Many studies provide little data with which to support their conclusions. They are often based upon anecdotal evidence or single-case observations, use comparisons of costs that represent only a small proportion of a software platform's total TCO (e.g., software purchase costs) and/or make simplified assumptions about operating conditions

(e.g., staffing costs will be identical for different software platforms) that provide a somewhat dubious foundation upon which to extrapolate to the costs that would be incurred by a broader population of firms.

To illustrate this problem, we divided the articles and studies that reported conclusions on TCO for a number of competing operating systems into four categories: Those that relied upon anecdotal evidence, price lists and/or informed speculation as the basis for comparison; those that provided actual data from a case study of only one particular firm or organization and the specific context it faced; those that used a database of survey returns to project the results for one particular firm or organization and the specific context it faced; and those that used data from a large sample of firms (i.e., greater than 10) to examine and report TCO across one or more generic types of workload. Note that in a few cases, it was difficult to classify a study into one of these categories. For our purposes, however, we are not so interested in the precise number of studies in each category, as the *pattern of distribution* across these categories.

Table 2. Many "TCO" Studies Provide Inadequate Data to Support Their Conclusions

Category	*Number of Articles/Studies*
No data: Anecdotes, price lists, and informed speculation	42
Case study data: In-depth figures on a *single* firm	1
Survey database: Used to project TCO for a specific firm	9
Survey sample: Used to report TCO for specific workloads	3

As we can see, despite the widespread interest in evaluating TCO, many of the articles and studies that weigh in on the debate actually bring very little *evidence* to bear on the topic. While the reporting of case study data provides some insight on outcomes, the fact that actual TCO figures vary widely across firms, as well as across specific workloads (see below) means that the value of such "single-point" studies is rather limited. Even small changes in a firm's context relative to the case study example are likely to render the comparison invalid from a TCO perspective. To provide meaningful conclusions in a TCO study, there is a need to ensure a much larger base of observations is used to give greater confidence that the costs actually observed are representative of a broader population of firms (and not an artifact of the specific firm chosen). In practice, this can be done either by collecting detailed surveys from firms over time, which can

then be used as the basis for estimating TCO for specific IT configurations (e.g., as is done by the Gartner Group) or by employing a large sample of respondents to capture data on more generic workload costs for a specific study (e.g., as has been done by IDC, in studies sponsored by both Red Hat and Microsoft).

2: There is little consistency in the method of calculating TCO across different studies.

Across all the articles and studies we reviewed, there is a notable lack of consistency in the methods for calculating TCO. These differences are apparent in both the costs that are included in an evaluation of a software platform's TCO, and the manner in which these costs are actually measured.

The main area in which studies differ is with respect to the number and type of costs that are included in a TCO calculation. At one end of the spectrum are studies that consider only one or two "direct" costs, focusing mainly on the up-front cost of the particular software platform that is being evaluated. At the other end are studies that use a much more sophisticated assessment, capturing not only the direct costs involved in purchasing and maintaining and supporting a particular platform, but also the "indirect" costs associated with the use of the system, such as the cost of downtime. Importantly, the more comprehensive studies typically find that the cost of purchasing or leasing the software represents only a small proportion of the TCO for a software platform (see below). This illustrates the danger of relying on studies that use only partial data on cost as the basis for making IT investment decisions.

The second area in which studies differ is in how they actually measure the costs that are part of the TCO calculation. These differences include differences in the timescale over which assets are depreciated (from one to five years in most studies), whether list or bid prices are used for software acquisition costs, how the costs of downtime are assessed (e.g., loss of salaried time versus a loss of business calculation) and the resulting metric that is used for a final comparison (e.g., cost per user, cost per processing unit, etc.). While most of the more comprehensive studies provide a description of how these choices are made in each case, the lack of consistency between them makes comparisons difficult. Indeed, it also raises the possibility that the particular results obtained were in fact an artifact of the specific choices made as to how to calculate these costs.

It is interesting to note that evaluations of TCO are not always conducted in a consistent manner even within the same firm. An example is provided by two recent studies performed by IDC: the first examined the TCO for Linux versus proprietary Unix systems (being funded by Red Hat, a vendor of products and services for Linux); the second examined the comparative TCO for Windows versus Linux (being funded by Microsoft). These studies varied on several important dimensions, including:

- The costs included as part of TCO. For example, in one study, the cost of system downtime was estimated which, on average, represented a significant 23.1% of total ownership costs; in the other study, this cost was not considered.
- The timescale over which costs were amortized. In one study, costs were aggregated into a one-year TCO; in the other study, costs were aggregated into a five-year TCO. Given some costs are one-off costs (e.g., purchase costs) whereas others are incurred annually (e.g., support costs) the use of different timescales will change the relative proportions of these costs in the final TCO figure.

It is important to emphasize that these comments are not meant as a criticism of IDC – in fact, the two studies cited are among the more comprehensive evaluations of TCO that have been conducted to date. The comments merely indicate that the lack of consistency in approaches to evaluating TCO exist both between and *within* firms doing the studies.

So why should we worry about the inconsistencies in evaluating TCO across different studies? The bottom line is that these inconsistencies can lead to very different conclusions in studies that are evaluating rather similar workloads. To illustrate, consider three recent studies that examined TCO comparisons for Linux, Windows and Unix software platforms when used for web-server/Internet/intranet workloads (two of the studies were performed by IDC, one by the Robert Frances Group). Of course, direct comparisons between these studies are made impossible by the absence of a consistent metric for measuring TCO, hence we must look instead at the percentage difference in cost identified *within* each study. Using Linux as the baseline platform, we observe that:

- The first IDC study found that Unix had a TCO that was 80% higher than Linux; by contrast, the Robert Frances Group found that Unix (specifically in this case, Solaris) had a TCO that was 650% higher than Linux.
- The second IDC study found that Windows had a TCO that was 6% higher than Linux; by contrast, the Robert Frances Group found that Windows had a TCO that was 150% higher than Linux.

Clearly, the differences in these evaluations are rather substantial. While the specific workloads and implementations examined in each case likely drive some of these differences, a large part of the variation probably lies in the methodologies used. Without greater consistency in these methods, we cannot be sure that any of these results are likely to hold in future studies, nor that we could replicate them if asked to actually implement the software platforms evaluated in a target organization.

It is important to note that even with a consistent approach to calculating TCO, evaluations of the costs of running similar workloads on the same software platform would still differ. This is because firms vary in the levels of reliability and availability they require from systems, the operating practices they adopt, and how close to "best practice" they perform in terms of efficiency. Critically, however, the observed variation in TCO would be an extremely useful piece of data *if calculated using a consistent methodology*. This data would help us to understand the confidence intervals for the costs that are likely to be achieved by a firm adopting a particular software platform for a specified workload. The problem at present stems from the fact that most of the variations in TCO that are reported by studies that examine the same workload are likely to come from differences in methodology, which end up swamping the *actual* differences in cost.

3: The data are biased by how organizations <u>currently</u> deploy their software platforms.

Many of the more comprehensive studies that evaluate TCO use data on the actual costs incurred by firms that presently operate the software platforms being examined. These data are used to extrapolate to an overall TCO figure for a generic type of workload. This methodology suffers from a specific type of bias, however, to the degree that firms use a particular vendor's software platform for highly specific, and less complex workloads. The problem is not so much that the data from such surveys are not accurate, as that these data are unlikely to prove reliable as a guide to the TCO that would be obtained if that same platform was used across a set of workloads with greater complexity.

To illustrate this bias, consider comments made in one study by IDC that compared the TCO of Linux with the TCO of proprietary Unix systems, using data obtained in telephone interviews with 142 managers (one of the highest sample sizes we

observed for a TCO study). The results show substantially higher staffing support costs for Unix versus Linux in both web serving and collaborative workloads (i.e., email, group calendaring and scheduling, shared folders/databases, etc.). Some might regard this as somewhat of a surprise, given the similarities between Linux and Unix. However, the authors note when discussing this result that "This is likely because of the higher incidence of multiple workloads aboard RISC/Unix and its generally more complex configuration (including ERP, database, and other complex workloads)." They add, "…larger Unix systems are more likely to be part of a more complex infrastructure (such as a cluster, or they are providing critical services to other servers)."

The implication of these comments is that in this particular study, Linux support costs may be biased lower in some cases because the software is currently used for less complex tasks (and in less complex configurations) than its Unix counterpart. That is, we do not have a true "apples to apples" comparison, because firms chose to use one vendor's software platform in a specific situation (i.e., one that it was most suited to). The net result is that despite the admirable efforts to collect a large amount of actual data on the deployment of these different software platforms, we are unsure of the extent to which Linux advantages in staffing support would be replicated if used in the more complex user scenarios that the Unix systems in this study typically addressed[1].

So what can be done about this potential bias? Here, we can only point to the need for a greater base of observations from firms providing actual data on costs in situations where different vendors' software platforms are used for a set of tasks (and in configurations) of similar complexity. Furthermore, assuming firms begin to gather TCO data in a more consistent fashion, the increasing amount of cumulative (and comparable) data that will become available should allow us to better estimate the impact of operating environment complexity on the cost of ownership, even without having a true apples to apples comparison. Until that time, firms conducting TCO studies should, at a minimum, provide some commentary on how managers' current choices of where to use particular software platforms might influence the ability to generalize results to environments of differing complexity.

[1] Note that this study also found substantial differences in software and hardware acquisition costs between these two software platforms. We limit our discussion to the differences in staffing support costs.

4: There is no accounting for differences in the "value" delivered to end-users.

Many TCO studies take no account of differences in the value delivered to users. The core assumption is that different software platforms provide essentially the same set of benefits to users, and therefore the primary selection criterion should be cost. This is not so much a critique of the methodology, as much as it is a commentary on the assumptions embedded in a TCO-based approach, versus an approach that might also attempt to capture differences in value delivered. These differences derive from, among other things, differences in product features and functionality, as well as the "option" value that comes from having a greater choice of applications for a particular software platform.

As an illustration, consider a TCO study that examined the costs of running Microsoft's Office productivity suite on a Windows platform as compared to the costs of running OpenOffice (an open source alternative) on a Linux platform. Specifically, the question arises as to how one should attach value to the large differences in functionality between Office and OpenOffice? Clearly, these differences are likely to be more important for certain *types* of user, illustrating the difficulty of performing such a calculation. Nevertheless, blindly applying a TCO-based logic without considering the question of value risks a flawed purchase decision. A further illustration of this point comes from considering the choices of application that are available to users in each case. Users of Windows may choose between using Office or OpenOffice, whereas users of Linux only have the option of using OpenOffice. In general, a greater choice of applications on a platform provides greater "option" value to users, a factor that is not captured in a typical TCO calculation.

Our argument suggests that managers considering investments in IT infrastructure must take account of the inherent strengths and weaknesses of a TCO-based metric. In particular, given the inability of such a metric to capture the differences in value delivered to users (at least, in the present forms that this metric takes) this factor must be evaluated separately to arrive at a robust decision. When we combine this insight with the notion that purchase decisions require both an evaluation of TCO, as well as a calculation of the "Return-on-Investment" from potential IT expenditures (e.g., to evaluate whether it is worth switching from one platform to another), we are left to

conclude that IT investment decisions must necessarily consider *three* different dimensions if they are to adequately characterize the costs and benefits of the various IT options facing a firm (see Figure 1).

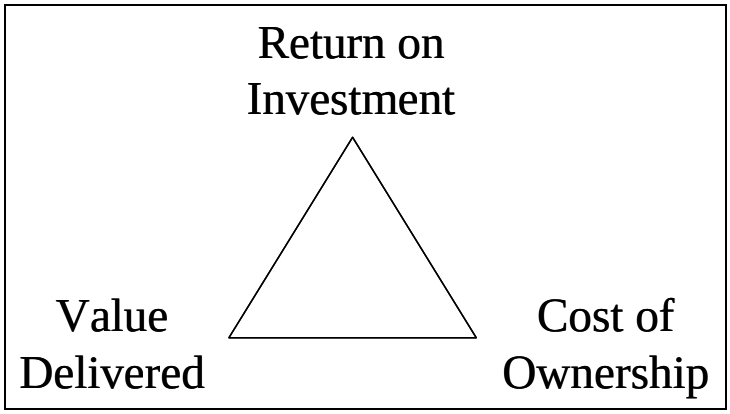

Figure 1. Performance Dimensions for Assessing IT Investments

What Conclusions Can We Draw from the Data?

Despite the inconsistent approaches to measuring TCO apparent in the articles and studies reviewed for this paper, there are still a number of general conclusions that can be drawn from the data assembled. We certainly don't take credit for reporting these conclusions for the first time – many organizations involved in evaluating TCO have done this already and, indeed, repeated them many times over. Nevertheless, given the discussion above, it is worthwhile to re-emphasize those conclusions that are robust.

1: Calculating the "true" TCO for a software platform is a complex activity.

The first conclusion that emerges from a review of existing studies is that calculating the true TCO for a software platform is a complex activity. There are many different types of costs that need to be captured, ranging from the up-front costs of purchase for the software and related hardware to the ongoing administrative costs incurred in support of the system over time. There are also many different ways to actually measure these costs, especially those "indirect" costs such as system downtime, which can be highly dependent on the context in which a system is used. As an illustration of the complexity involved in a robust TCO calculation, consider that Gartner Group's "Chart of Accounts," which describes the costs it captures in its proprietary TCO

database, is 27 pages long. Gartner appears to have realized the value in making sure its methodology is both *comprehensive* and *consistent* in the data it captures.

2: TCO results are highly dependent on the particular workloads being evaluated.

The next conclusion that emerges from reviewing existing studies, and in particular, those that examine different types of workload using the same methodology, is that the results are highly dependent on the particular characteristics of each workload. Indeed, variations in TCO across different workloads (e.g., web server, file server, print server, etc.) are often much greater than the variations in TCO across different vendors for the same workload. To illustrate, consider results from an IDC study that compared Linux and Windows across five different workloads. It is noticeable that the TCO figures vary by almost an order of magnitude across workloads, whereas the difference between Linux and Windows is typically much smaller – between 6% and 23% over a five-year period.

Table 3. Different Server Workloads Drive Great Variations in TCO ($ per 100 users)

Application	*Windows TCO*	*Linux TCO*	*Win/Lin Ratio*
Networking	11,787	13,263	0.89
File	99,048	114,381	0.87
Print	86,849	106,989	0.81
Web	32,305	30,600	1.06
Security	70,495	90,975	0.77

Source: *Windows 2000 versus Linux in Enterprise Computing,* IDC, 2002

3: Software acquisition costs are dwarfed by other costs.

A review of the more comprehensive studies of software TCO illustrates that all tend to agree on one conclusion. Software acquisition (or lease) costs are somewhat unimportant in terms of their overall impact on TCO. This is a particularly important fact, given the recent rise to prominence of open source or "free" software, which can be purchased at a comparatively low price and copied thereafter without additional cost. Some observers have argued that this fact alone dictates that such software will have a lower TCO. The results from most TCO studies, however, suggest that such a conclusion is wrong. On the contrary, these studies suggest that whether software is free, cheap or relatively expensive will often have little impact on the outcome of an investment

decision for a software platform, given that other costs typically dominate the TCO calculation.

To illustrate, consider the following results from studies that compared the TCO for different proprietary operating systems versus Linux on a variety of workloads. We do not report software costs for Linux, given these are, by definition, lower than the ones we report here. It is noticeable that in all but one case, the purchase cost of the software is less than 10% of the total platform TCO. When one realizes that this cost includes both the operating system software as well as other software specific to the workload, it is evident that the purchase costs attributable to the operating system alone are unlikely to be a major factor in IT investment decisions.

Table 4. Software Purchase Costs Represent Only a Small Fraction of TCO

Study	*Workload*	*Vendor*	*Software cost as % of Total TCO*[2]
Robert Frances Group	Web Server	Solaris	4.9%
Robert Frances Group	Web Server	Windows	4.2%
IDC	Networking	Windows	1.8%
IDC	File	Windows	4.0%
IDC	Print	Windows	1.9%
IDC	Web Server	Windows	22.0%
IDC	Security	Windows	8.3%

Source: *Total Cost of Ownership for Linux in the Enterprise*, Robert Frances Group, July 2002; *Windows 2000 versus Linux in Enterprise Computing*, IDC, 2002.

4: Staffing costs are often the single largest component of TCO.

The largest individual category of cost identified in most comprehensive evaluations of TCO is staffing. This category typically includes the costs of supporting the software platform being examined, as well as the costs of system administration. To illustrate the proportion of costs staffing represents, consider the results from studies by two different organizations looking at the differences in TCO for Linux versus Windows[3].

[2] Note that the time horizon used for these studies differed (one used three years, one used five years) hence the proportion of costs represented by software are not directly comparable between these studies. Note also that the IDC study included training, installation and maintenance in this category.

[3] Note that for proprietary Unix systems, some comparisons show staffing is second to hardware in terms of cost, given the more expensive non-Intel based hardware that is used for these systems.

Table 5. Staffing Is Often the Largest Part of TCO

Study	*Workload*	*Vendor*	*Staffing cost as % of Total TCO*[4]
Robert Frances Group	Web Server	Linux	48.4%
Robert Frances Group	Web Server	Windows	75.3%
IDC	Networking	Linux	61.8%
IDC	Networking	Windows	71.1%
IDC	File	Linux	71.0%
IDC	File	Windows	54.5%
IDC	Print	Linux	55.2%
IDC	Print	Windows	46.3%
IDC	Security	Windows	78.1%
IDC	Security	Windows	71.8%
IDC	Web Server	Linux	75.2%
IDC	Web Server	Windows	46.7%

Source: *Total Cost of Ownership for Linux in the Enterprise*, Robert Frances Group, July 2002; *Windows 2000 versus Linux in Enterprise Computing*, IDC, 2002.

5: Staffing costs are driven by many factors, including functionality and availability.

Given the two results above, we can see that rather small differences in staffing cost are likely to overwhelm differences in software acquisition cost. In particular, given that in the more comprehensive studies of TCO, staffing represents over 60% of costs, a difference of only 10% in staffing costs between platforms will often outweigh any advantage that might come from cheaper software. This result suggests that studies of TCO would be best served by digging into much greater depth on the drivers of differences in staffing cost. Getting this right is *much* more important in the grand scheme of things, than making sure you have adequately captured any discounts that will be received on the software's purchase price.

Studies that provide greater detail on staffing costs for specific software platforms demonstrate that a complex mix of factors drives this cost. In particular, the specifics of the workload that is being examined will have a large influence on staffing, given that certain workloads put much greater strain on a system in terms of administration. Indeed, the TCO evaluations across workloads cited earlier back up this conclusion. When comparing similar workloads however, the next factors that are typically considered

[4] Note that the time horizon used for these studies differed (one used three years, one used five years) hence the proportion of costs represented by software are not directly comparable between these studies. Note

include both the functionality that is included in the software to aid system administrators, as well as the requirements of the system in terms of downtime.[5]

With regard to the former, many software platforms are at different stages of maturity in terms of the functionality they provide to system administrators, the system management tools that are available for the platform, and hence the staffing costs that are required to achieve similar levels of performance. While this topic is a moving target – all vendors are in general, moving to develop greater functionality that ultimately reduces staffing requirements – the differences that exist between platforms are likely to remain a primary driver of TCO and hence platform attractiveness for as long as staffing remains a major cost. With regard to the latter, it is noticeable that different software platforms vary in the sophistication of the tools that are provided for managing availability. In many studies, the differences in such tools are found to be more critical in assessing downtime than differences in the native reliability of the underlying software. This is because downtime tends to be attributable more to the demands placed on a system by the applications that provide services to users, not the operating system itself (e.g., print and file server workloads tend to have much greater downtime than other workloads, regardless of the operating system). In combination with the discussion above, we are left to conclude that the evaluation of staffing costs – the largest component of TCO for a software platform – is likely to depend to a significant degree on differences in the functionality and tools available to software platform users to help them manage workloads of interest. Studies of TCO for different software platforms would be well advised to place the bulk of their methodological efforts to developing better ways of assessing how such differences are likely to impact users.

Conclusion

Despite the recent interest in comparing the TCO for different software platforms, we find that the state of practice for evaluating TCO is rather poor. At one end of the spectrum, some articles on the topic provide little more than price lists of competing

---

also that the IDC study included training, installation and maintenance in this category.

[5] Some studies capture the costs of downtime separately, which in general is a welcome practice. Here, we consider this cost only in terms of the additional staffing required to maintain a specified level of system availability.

vendor's products, which as noted earlier, often represent no more than 10% of the "true" TCO for a software platform. Others provide data on only single case observations which, given the heterogeneity apparent in TCO outcomes across firms, are clearly inadequate as a basis for drawing more general conclusions. At the other end of the spectrum, some organizations have made commendable efforts to develop comprehensive and robust methodologies that make use of larger samples of data, thus providing greater confidence on estimates of TCO (e.g., Gartner and IDC). However, there is still a considerable amount of inconsistency in the metrics used to report TCO, the timescales over which costs are considered, the costs that are included in these evaluations, and the manner in which these costs are calculated. Importantly, these inconsistencies exist both *between and within* the different firms that conduct these evaluations.

The evidence suggests a clear need for a set of more universal guidelines on how TCO evaluations should be conducted, thereby allowing comparisons to be made between different studies conducted by different firms. Only once this has been done will we have developed a foundation for making the study of software platform total cost of ownership a cumulative science, rather than its present state, in which a series of isolated, disconnected facts are reported, many of which conflict with each other.

At a broader level, our work suggests a need to realize the limitations of a TCO-based metric, given this represents only one dimension of the IT investment problem. While some consulting firms have identified the need to provide a Return on Investment calculation alongside the evaluation of software platform TCO, we highlight a third dimension that is missing from most analyses. Specifically, software platforms often differ in the *benefits delivered to users*, these differences being driven by variations in the functionality embedded in products, or the differing set of choices that are available to users in terms of the applications that each platform supports. We conclude that any evaluation of alternative platform investments must therefore consider *three* important dimensions to ensure a robust decision is made; the total cost of ownership, the value delivered to users, and the return on investment achieved by planned IT expenditures.

Appendix: List of Articles and Studies Reviewed by Category

Category: No data: anecdotes, price lists, and informed speculation

Mike Berman, OsOpinion.com **December 5, 2002**
The Linux Cost Controversy

Joe "Zonker" Brockmeier, OsOpinion.com **November 12, 2002**
Is Windows Cheaper Than Linux?

CIOview **2001**
The Financial Impact of Migrating to Linux

Nancy Cohen, Open
TEI and Sympathy

Cybersource **2002**
Linux vs. Windows: Total Cost of Ownership Comparison

Fortuitous Technologies **2000**
The Benefits of Linux

Michael Gartenberg, Computerworld **October 30, 2000**
Myths behind TCO

Gartner **October 15, 2002**
Explaining Gartner's TCO Manager Software

Gartner **June 19, 2002**
How to Avoid Pitfalls and Save Money with Linux Servers

Gartner **March 20, 2002**
Software Licensing Implications of Server Consolidation

Gartner **March 2, 2000**
TCO Manager: A Minor Player in Server Platform Decisions

Gartner **June 8, 2001**
The Price of Information Security

Gartner **October 16, 2002**
The Total Cost of Ownership Index: Defining the Database

Gartner **June 19, 2001**
When and When Not to Consider Linux on the Mainframe

Gartner, CNET.com **November 1, 2001**
Commentary: Linux's Murky Prospects

Grant Gross, TechRepublic **December 3, 2002**
Linux's TCO: Not Free, but Still Impressive

Mark Hall, Computerworld **May 13, 2002**
TCO: Linux Delivers on Big Iron

Richard Hillesley, LinuxUser **May 2001**
Money to Burn?

Thomas Hoffman, Computerworld **December 2, 2002**
TCO: Flawed but Useful

Rob Lineweaver **November 21, 2002**
Cost Savings of Open Source Software in the Server Room: An Informal Case Study in K-12 Education

Vance McCarthy, IDN **November 18, 2002**
Gartner Warns Linux Can Cost More to Integrate

Scott McCollum, WorldTechTribune **November 22, 2002**
Gartner Group: Linux TCO Savings Misleading, Costs More to Integrate

META Group **November 22, 2002**
Linux Servers: No "Silver Bullet" for Total Cost of Ownership

META Group **November 15, 2002**
Reducing TCO: Going Beyond Server Consolidation

META Group **December 2, 2002**
The Real Value of Linux

MITRE **2001**
The Value of Return on Investment (ROI) Analysis to the Government and Other Non-Profit Organizations

Paul Murphy, LinuxWorld **October 18, 2001**
A Strategic Comparison of Windows vs. Unix: The Real Question Is Not "Which Is Cheaper" but "Which Is Smarter?"

Paul Murphy, LinuxWorld
RE: IDC Declares Linux 80% Cheaper Than Unix

Paul Murphy, LinuxWorld **October 18, 2002**
The Pros and Cons of Business-app Implementation Via Open-source Software (Part 1)

Paul Murphy, LinuxWorld **April 3, 2002**
Total Cost of Ownership Series Revisited

Paul Murphy, LinuxWorld **February 1, 2002**
Virtual Case Study: Betting on Linux at Andersen Printing

Paul Murphy, LinuxWorld **November 16, 2001**
Virtual Case Study: Cutting IT Costs with Unix

Paul Murphy, LinuxWorld **January 4, 2002**
Virtual Case Study: How an Airline Can Find Efficiency with Unix

Paul Murphy, LinuxWorld **December 4, 2001**
Virtual Case Study: Saving a Small Software Developer with Linux

Paul Murphy, LinuxWorld **February 19, 2002**
Virtual Case Study: Systems Decisions at Cutter Mills

Thomas Pfau **October 29, 1999**
An Analysis of Microsoft's TCO Comparison—Part 1

Brendan Scott **July 15, 2002**
Why Free Software's Long Run TCO Must Be Lower

Patricia C. Skarulis & A. Jerome York **September 8, 1997**
Enterprise Desktop Management

Jan Stafford et al, SearchWindows-Manageability **August 15, 2002**
Linux-Windows TCO Contest "a Wash" for Now

UT-Austin Business School **May 10, 2000**
A Comparative Analysis of Red Hat Linux 6.1 and Windows 2000 Server

Jaikumar Vijayan, Computerworld **June 18, 2001**
The New TCO Metric

David A. Wheeler **November 24, 2002**
Why Open Source Software / Free Software (OSS/FS)? Look at the Numbers!

Category: Case study data: in-depth figures on a single firm

Microsoft
Hosting Company: The Hidden Costs of Using Linux: A Cost Analysis Between Red Hat Linux and Windows 2000 Server

Category: Survey database: used to project TCO for a specific firm

ConsultingTimes **2001**
The Truth Behind the Great Server Heist: ConsultingTimes Does the Math

Equant
Equant TCO Study Comparing Microsoft Exchange Server on NT with Bynari Trade Server Running on Linux for IBM S/390

Gartner, sponsored by Microsoft **August 2001**
Report/White Paper: IT Cost Analysis in the Public Sector in Germany

IBM **April 2002**
Bynari eDocuments: Rich Lechner: Consolidating NT/Exchange with a Linux Mainframe

Microsoft **December 2, 2002**
An Operating System Even a CFO Would Love: The Business Value of Windows 2000 Server

Microsoft **January 2002**
Gartner TCO Study Summary January 2002

Microsoft **November 22, 2001**
Gartner TCO-Tutkimus: Summary of TCO Analysis for the Town of Vaasa

Microsoft **July 2002**
The Business Value of Upgrading to Windows XP and Office XP: Windows XP Beats Linux in TCO Studies in Finland: Two Cities Show 36-38% Lower TCO with Windows XP

Standish Group **2002**
Dollars to CENTS: TCO in the Trenches 2002

Category: Survey sample: used to report TCO for specific workloads

IDC, sponsored by Microsoft **October 2002**
Windows 2000 Versus Linux in Enterprise Computing

IDC, sponsored by Red Hat **November 2001**
The Role of Linux in Reducing the Cost of Enterprise Computing

Robert Frances Group **July 2002**
Total Cost of Ownership for Linux in the Enterprise